AFRL-VA-WP-TP-2002-312

APPLICATIONS OF NEURAL NETWORKS IN FLIGHT AND SYSTEM CONTROL TECHNOLOGY

Todd T. W. Bruner

JUNE 2002

AIR VEHICLES DIRECTORATE
AIR FORCE RESEARCH LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7542

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
June 2002	Conference Paper Preprint	03/26/1998 – 12/31/2001

4. TITLE AND SUBTITLE
APPLICATIONS OF NEURAL NETWORKS IN FLIGHT AND SYSTEM CONTROL TECHNOLOGY

5a. CONTRACT NUMBER
F33615-98-C-3603

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
65502F

6. AUTHOR(S)
Todd T. W. Bruner

5d. PROJECT NUMBER
3005

5e. TASK NUMBER
30

5f. WORK UNIT NUMBER
9E

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Digital System Resources, Inc. (DSR)
12450 Fair Lakes Circle, Ste 500
Fairfax, VA, 22033

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Vehicles Directorate
Air Force Research Laboratory
Air Force Materiel Command
Wright-Patterson Air Force Base, OH 45433-7542

10. SPONSORING/MONITORING AGENCY ACRONYM(S)
AFRL/VACC

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)
AFRL-VA-WP-TP-2002-312

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
Proceedings for AUVSI Unmanned Systems 2002 Conference.

14. ABSTRACT *(Maximum 200 Words)*
This paper discusses research Digital System Resources, Inc. (DSR) performed in an application of neural network technologies for flight control.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT:	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON (Monitor)
a. REPORT	b. ABSTRACT	c. THIS PAGE			Thomas Molnar
Unclassified	Unclassified	Unclassified	SAR	22	19b. TELEPHONE NUMBER *(Include Area Code)* (937) 255-8439

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18

Application of Neural Networks in Flight and System Control Technology

By Todd T. W. Bruner, Digital System Resources, Inc. (DSR)
(703) 418-9176, tbruner@dsrnet.com, www.dsrnet.com

1 BACKGROUND

Figure 1 Miniature Air Launched Vehicle

This paper discusses research Digital System Resources, Inc, (DSR) performed in an application of neural network technologies for flight control. DSR under Small Business Innovative Research (SBIR) contracts with the Air Force Research Laboratory (AFRL) at Wright Patterson Air Force Base has successfully designed pilot behavior neural networks and algorithms for autonomous vehicles. We have succeeded in development of autonomous control techniques, algorithms and tactical decision methods. These networks, algorithms and methods have been demonstrated in both a simulation environment platform and a Miniature Air Launched Vehicle (MALV) platform (Figure 1). Testing was verified and validated in the simulation program and hardware-in-the-loop (HITL) testing giving the software a technology readiness level (TRL) 4/5. DSR has at least five years of experience working with AFRL in developing the concept, developing the algorithms, and finally applying the algorithms to a flight vehicle. Initially DSR was awarded a contract to develop the concept of autonomous flight control by use of neural networks. AFRL second contract with DSR was to develop UAV Flight Control Technologies algorithms and neural networks to:

- Design and demonstrate sophisticated neural networks to perform select functions of a simulated autonomous vehicle.
- Prove, via simulation, the ability of multiple neural networks to continuously perform their functions in parallel, with the collective result being the safe execution of a mission by a constellation of four autonomous vehicles operating in formation.
- Prove the ability of neural networks to perform conflict resolution between neural networks.

After DSR proved successful in a simulation environment, the company was contracted to apply these algorithms to DARPA's MALV program. DSR was fortunate to work with Northrop Grumman Ryan Aeronautical Center. Their expertise and willingness to apply theses advance concepts was the enabling activity and success in our testing and progress through HITL testing.

This paper addresses:

- The neural network development process
- Demonstration of selected neural network flight control technologies
- The lessons learned from the interfacing of the neural networks to flight vehicle
- Future applications in unmanned systems and support systems

2 THE NEURAL NETWORK DEVELOPMENT PROCESS

DSR utilized neural networks for several reasons. First, neural networks can be used for flight control functions, which are difficult to perform with traditional approaches. Neural networks are effective at solving problems where the relationships between the input variables and output variables are not well understood; traditional algorithms, and if-then expert system rules are irrelevant to these problems. Second, neural networks are mature and well-understood technology, especially the feed-forward neural networks used in this architecture. Third, the feed-forward neural networks cannot alter their training after being placed in a platform, so their behavior is predictable and reliable. For example, if a friendly F-16 mistakenly shoots at a UAV, the UAV will not learn that F-16s are the enemy.

Function Determination: Determining which functions that need to be performed is the first step in our process. Our neural networks are involved in a number of functions: getting the platform to the area of operations; determining the validity of a preprogrammed or detected target; determining the need to launch a weapon against that target; making the decision to fire the weapon from the "best" unit in the constellation and assigning that unit to fire; and ensuring that the platform is in the correct location to launch that weapon. DSR's neural networks tell the platform WHAT TO DO, not HOW TO

DO IT. The latter is a function of the platform flight control or mission computer. This is an important distinction. This defines the limitation of our system. The neural networks do not replace existing software algorithms used for flight control functions or navigation in current generation unmanned air vehicles (UAVs). What they do is significantly expand the range of scenarios that the flight control/mission computer is able to correctly and reliably respond to. They add a level of intelligence to the platform, enabling it to recognize and react appropriately to any situation, with the goal of dramatically reducing the level of operator intervention.

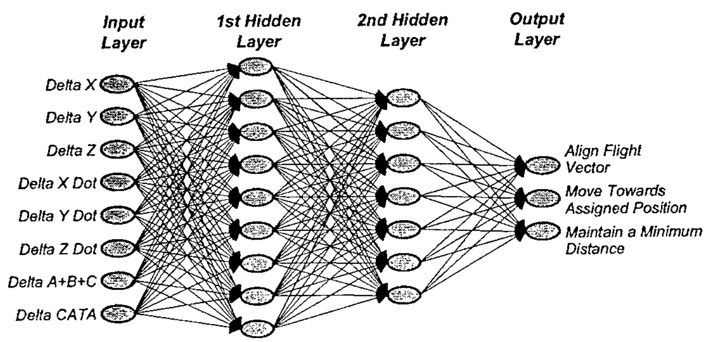

Figure 2 Sample Neural Network Diagram

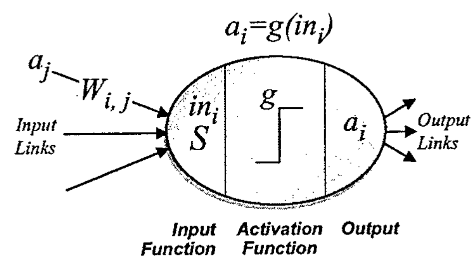

Figure 3 Sigmoid Function

Neural Network Node: We use the perceptron model (Rosenblatt, 1958)[1]. The network consists of layers of nodes and the weighted connections between the nodes. Outputs are produced at the output layer by following several steps. . The chosen output is the behavior that the UAV should perform at the current time. Each node value is created by a two-step algorithm, in this case. Some of our networks are single step. There are many possibilities for activation functions, such as a step or sigmoid function (figure 3). The result is a hybrid fuzzy neural net.[2] DSR selected the feed-forward neural network type to implement the UAV flight control subsystem. The feed-forward neural network

[1] Behavior-Based Robotics, Ronald C. Arkin, MIT Press, 1998.

[2] Intelligent Control: Principles, Techniques and Applications, Zi-Xing Cai, World Scientific 1997

learns during laboratory training but does not learn after it is placed in a simulated or real platform. The deterministic nature of feed-forward neural networks is key for our applications with a UAV.

The next step is the design of neural network architecture. The number of input and output nodes is fixed according to the system; however, selecting the optimal number of hidden nodes is a difficult design issue, that is far from obvious, and often considered to be problem dependent. Intuition suggests that 'more is better' but this is not always our case. The number of hidden units and layers control the power of our model to perform the mission, but there is an associated trade-off between training time and model performance. Large hidden layers could also become counterproductive, as an excessive number of free parameters will encourage over fitting of training data, reducing the generalization capabilities of our network. We used no general procedure to determine the optimum number of nodes or layers. Trial and error, using different architectures and fixed stopping conditions, was our best approach. The number of layers in our neural networks and their connection structure greatly affects their performance. Usually a fully connected multiplayer network produces more accurate outputs but with a higher computational burden. In our case, since the network is to be integrated into another system, which is linked to computer operations, we selected a fully connected three or four layer neural network, consisting of input layer, one or two hidden layers and a output layer to balance accuracy and computational requirements.

Behavior Matrix: Our third step in the process was to develop the behaviors. DSR used a behavior-based methodology[3] to create intelligent agents using neural networks. This methodology asserts that a complex behavior can be produced through the interaction of a few simple behaviors. The neural network chooses the simple behavior that the platform executes in real-time. Over the course of time, the neural network produces a stream of simple behaviors. This stream of simple behaviors results in complex behavior. For example, the formation behavior consists of three simple behaviors, which are

[3] Behavior-Based Robotics, Ronald C. Arkin, MIT Press, 1998.

"Align Flight Vector", "Move Towards Assigned Position", and "Maintain A Minimum Distance". Suppose a UAV has finished pursuing a target and now wishes to rejoin its formation. The formation neural network repeatedly outputs "Move Towards Assigned Position" until the UAV is close to its assigned position, and then it repeatedly outputs "Align Flight Vector" to align the UAV with the rest of the formation. If at any time the UAV might collide with an aircraft or terrain feature, then the formation neural network outputs "Maintain A Minimum Distance". These three simple behaviors, performed at the right time, produce the formation behavior. The construction of any behavior matrix defines the inputs, outputs and the intersecting logic. The domain expert creates the logic for each intersecting cell. A description of this for the formation neural network follows.

Formation Neural Network Inputs: The formation neural network contains eight inputs. The following chart lists the inputs and also describes each of the inputs in detail.

INPUT	Description
delta X	Distance in feet from the UAV to the Virtual Lead along the direction the lead is flying
delta Y	Distance in feet from the UAV to the Virtual Lead along the direction of lead's wingtips
delta Z	Altitude difference in feet between the UAV and the Virtual Lead
delta speed	Difference in knots between the speed of the UAV and the speed of the Virtual Lead
delta heading	Difference in degrees between the heading of the UAV and the heading of the Virtual Lead
delta pitch	Difference in degrees between the pitch of the UAV and the pitch of the Virtual Lead
delta A+B+C	Distance in feet from the UAV to the closest UAV in the formation
delta CATA	Collision angle in degrees to the closest UAV

Formation Neural Network Outputs: The formation neural network contains three outputs: (1) Move to Assigned Position; (2) Align Flight Vector; and (3) Maintain Minimal Distance. The output "decision" that achieves the highest score during each neural network process is the chosen output that will be executed. The output "decision" is sent to the flight/mission computer where the actions are then executed.

Output	Description
Align Flight Vector	Align flight vector to that of the Virtual Lead
Move Towards Assigned Position	Move to assigned formation position
Maintain a Minimal Distance	Perform a series of maneuvers to avoid a collision

Formation Behavior Matrix: Figure 4 is the formation behavior matrix designed by a domain expert. The chart shows the interactions between the inputs and outputs of the formation neural network. The behavior matrix is converted to software code. DSR developed a neural toolkit which constructs the networks based on the network architecture chosen, reads the behavior matrix, develops training set data and trains the neural network. With an independent set of training data, the networks are tested. Once this is completed, then the networks are integrated into the system.

	Align Flight Vector	**Move Toward Assigned Position**	**Maintain a Minimum Distance**
Delta X		>200 ft =00 500 ft = 20 >2000 ft = 50	
Delta Y		>200 ft =00 1000 ft = 20 >2000 ft = 100	
Delta Z		>200 ft =00 1000 ft = 20 >2000 ft = 50	
Delta Speed	>5 knots =00 10 knots = 20 >20 knots = 40	>5 knots =10 10 knots = 20 >20 knots = 40	
Delta Heading	>2 degrees = 00 10 degrees = 20 >20 degrees = 40		
Delta Pitch	>5 degrees = 00 10 degrees = 20 >20 degrees = 40		
Delta A+B+C			>500 ft =200 1000 ft = 100 >6000 ft = 00
Delta CATA	>2 degrees = 25		>0 degrees = 100 10 degrees = 50 >30 degrees = 00

Figure 4 Formation Behavior Matrixes

This process was followed for the creation of the neural networks for both UAV and MALV. Five neural networks Formation, Navigation, Coordinated Engagement Weapon Control (CEWC), and Conflict Resolution were developed. Air To Ground, Formation and Navigation were integrated into the MALV vehicle. Listed below are the individual neural networks and the functions performed. Figure 5 illustrates the C2 Subsystem that illustrates the relationship between the modules and the subsystem.

FORMATION	NAVIGATION
Perform join-up of UAVs into a desired formation Maintain formation orientation through course and speed changes Change formation shapes Perform Collision avoidance	Waypoint Guidance Time Management Threat Avoidance Pop-up threats
CEWC	**AIR TO GROUND**
Resolve which UAV will shoot the target	Determine whether the UAVs release weapons
CONFLICT RESOLUTION	
Resolves conflicts between competing behaviors	

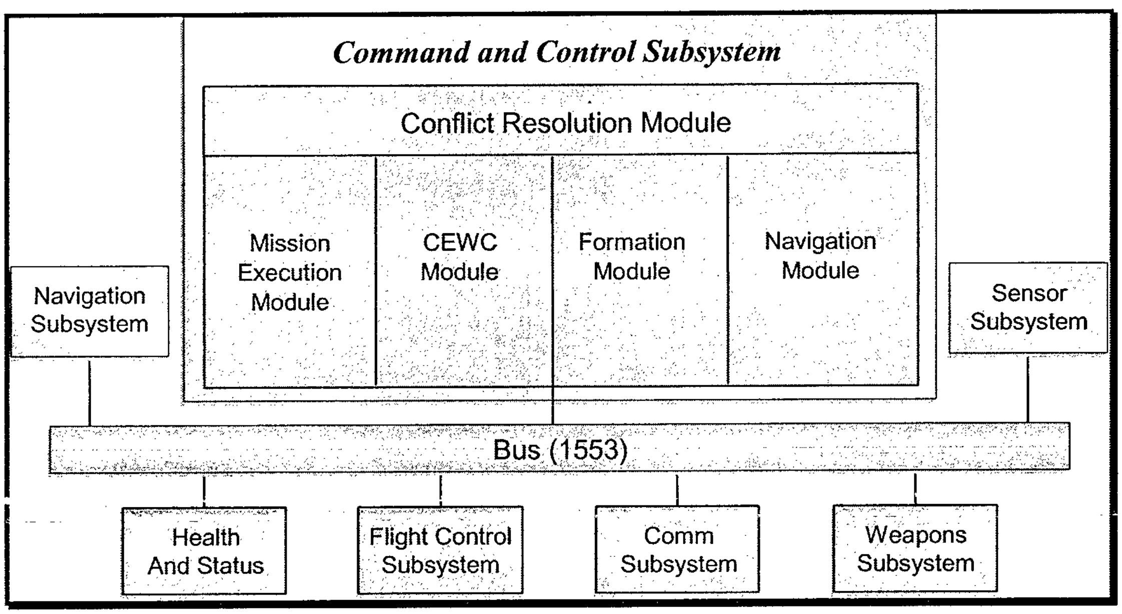

Figure 5 Command and Control Subsystem

In addition to the neural networks, algorithms were also developed by DSR to perform target engagement, cooperative target search, threat detection, and formation optimization.

Scenario Toolkit and Generation Environment (STAGE), a commercially available software application used to build scenarios in real-time synthetic environments, was used as a simulation environment to test their neural network and algorithm technology. So that the software developed could be portable to other systems, DSR also used its own Multipurpose Transportable Middleware (MTM). This proved invaluable in transitioning from UAV to MALV from a software production perspective, not design. MTM is an insulation layer of software that protects application programs from the nuances of different target hardware and operating systems. MTM protects the investment in application software when target hardware and operating systems change. The resultant UAV system architecture for the flight control technologies is depicted in Figure 6. This architecture provides a basis for further advancements.

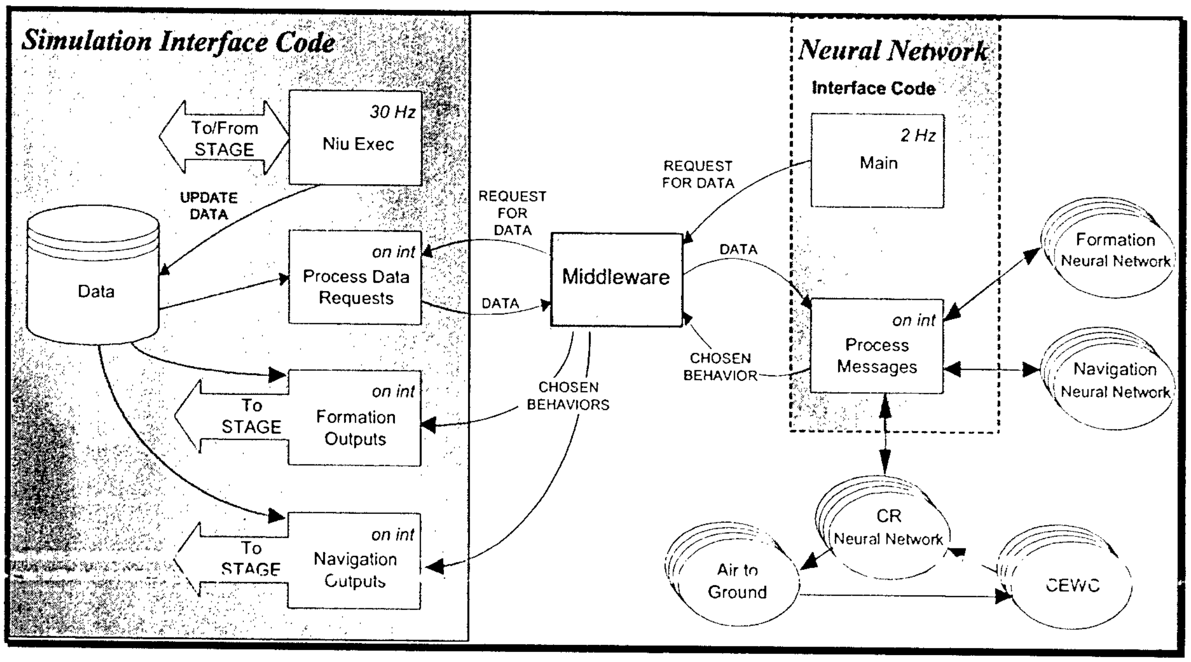

Figure 6 System Design - UAV

For MALV, DSR applied UAV lessons learned and redesigned the software to incorporate a more modular architecture. The new architecture provided DSR the ability to test the same code on STAGE, Ryan Aeronautical's All Software Simulation, and the HITL. This portability to any vehicle gave genesis to the name Intelligent Vehicle Autonomous Network (IVAN) Software (Figure 7).

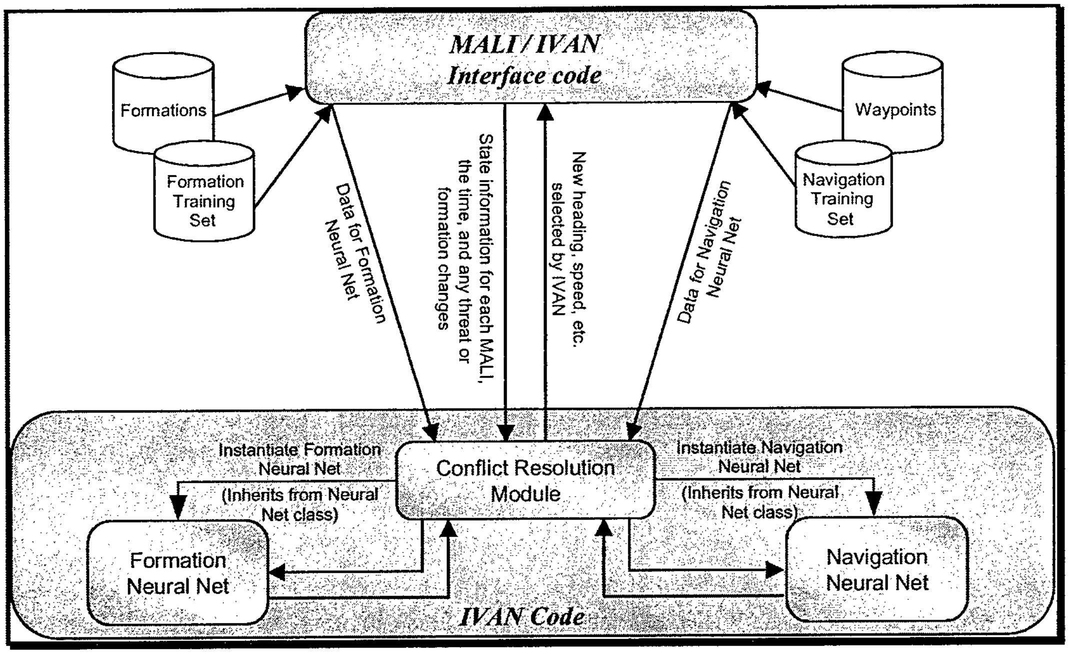

Figure 7 System Design - MALV

3 Demonstration

DSR has successfully demonstrated formation flying, join-up, cruise, target engagement, threat detection, threat avoidance, and collision avoidance in a simulated or HITL environments. DSR was able to achieve these successes by including qualified Air Force and Navy pilots from the beginning in the requirements, design and development process. Their extensive research in the objectives necessary to achieve mission effectiveness in an unmanned system were used as the basis for the algorithms and neural networks. These pilots are what are called in the neural net development process the "domain experts". They articulate the behavior of a pilot into a behavior matrix. The domain experts (pilots) then validated the behavior of the UAV operating by the algorithms.

4 Lessons Learned

There were several lessons learned throughout the development of the UAV and MALI programs. At the beginning of development DSR chose a structured coding approach that relied heavily on the simulation environment STAGE. Once awarded the MALI contract DSR soon realized an object oriented approach was better suited for the program. New functions were also written allowing the code to be independent of the simulation environment. The code was now easily transportable between STAGE and Northrop Grumman's All Software Simulation.

DSR also developed it's expertise in Neural Networks over time and realized that the training of the networks could be accomplished in a matter of minutes compared to the long hours of training that achieved the same results.

Domain experts developed their skills during the growth of the program. In the beginning the process from the behavior matrix to a complete running neural network could take several months and many iterations. With the understanding of the weights and the interaction between the inputs and the outputs of the neural networks the process time was shortened and less iterations of the behavior matrix were needed.

5 FUTURE APPLICATIONS

The ability of neural networks to be trained to replicate decisions of an expert will provide innovation to the way numerous industries and agencies function today, both within the Federal Government and the civilian business community. Any system that performs repetitive actions, responds to external stimuli, and is subject to minor variations is a candidate for the application of neural networks. Neural networks will enable economy of operation by responding to external stimuli and minor variations to the system it is monitoring. The neural network will adjust the system to maintain an optimal operating or design condition. The ability of neural networks to be trained to react repeatedly and reliably to specific circumstance produces very meaningful cost savings in both economy of operation and system life cycle cost. In terms of economy of operation, neural networks will enable functions to be executed using significantly less lines of code. The "If...Then" statements we are so accustomed to, will be replaced. The operations are based on a number of variables, and as the neural network is trained from the stimuli and variances received as inputs, the need for specific lines of code are reduced, adjusting the system to a more efficient state or design. The efficiency of the neural network can be used to free up capacity on computer systems for use by other applications. Neural networks have been designed for decision making – to take knowledge, apply it to the current situation, and refine its decision baseline as more knowledge becomes available, either from own ship sensors or from external sources.

Combined Technologies: DSR's latest research involving real-time flight control concept for a semi-autonomous/autonomous vehicle involved teaming with the United States Naval Academy (USNA) to combine our behavioral based deterministic neural network technologies with USNA's advanced techniques for controlling the null space permitting a decentralized control methodology, and the use of nonlinear optimization feedback control methods to control complex nonlinear system. Typically, autonomous control of vehicles implement either a behavior based approach or systems based approach. A systems approach provides a deterministic result, and generally requires significant algorithms and inter communications. In the case of lethal weapons, control systems typically demand such an approach. While a system approach provides provable results, typically this approach requires approximations to complex dynamics and relatively significant processing time. Behavior based systems demonstrate exceptional real-time performance and dynamic adaptability and robustness. But these, too, have their drawbacks, in that, generally, these systems do not use an explicit mathematical model approach or present a closed design. Our new concept combines these two approaches with recently developed algorithms. Consider the system architecture in Figure 8.

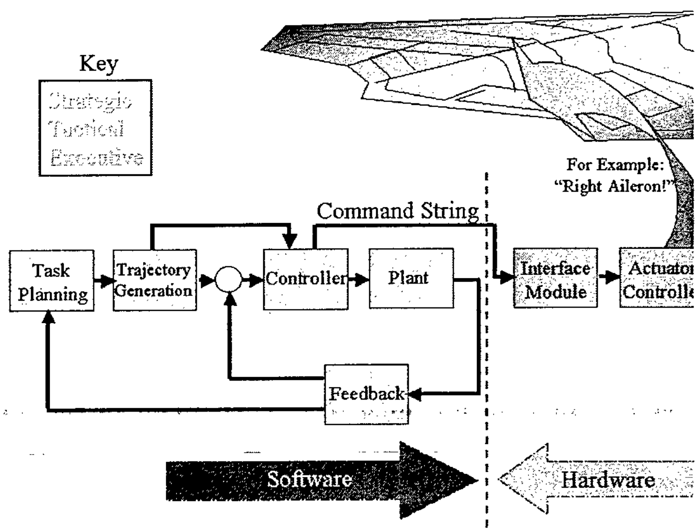

Figure 8 System Architecture

The Task Planning module can account for unknown terrain obstructions and moving threats such as radar, jammers, and other aircraft, by use of our neural networks to provide *operational* and/or *strategic* information to the trajectory generation scheme for real-time instantiation. The trajectory generation, control, plant and feedback modules perform in real time, optimally in the nonlinear region of a flight envelope. Trajectory generation expands the effort and offers flexibility for neural network integration with well-established system-theoretic robot control methods through application of redundant manipulator techniques. The plant may include dynamics of multiple vehicles, and the control may be centralized or decentralized. Feedback and control modules provide a model independent controller having the potential to control a variety of vehicle platforms within formation or platoons that are being tasked. The software architecture of Figure 8 incorporates three advanced techniques. These techniques include neural networks, redundant manipulator techniques for decentralized control, and feedback controller using dynamic quasi-Newton method based on non-linear least-squares optimization methods.

Control Of UAV Cooperative Behavior: Our current research has identified design issues associated with control of UAV cooperative behavior, to include control structure architectures and minimum data rates are depicted in Figure 9. Initially the constellation of four UAVs is sent out with a mission, which is achieved by use of their group behavior. As the constellation moves to a way point the formation changes. Was the formation changed due to reactive or deliberative control systems? Did this require a communication transmission? The constellation continues and one of the four UAVs is shot down. The constellation has the same reconnaissance mission of four targets. Is another UAV called up from reserve? Is the unassigned target left alone? Do the remaining three UAVs retask themselves? Is the man-in-the-loop required to transmit a new mission? Ideally, the remaining three UAVs would think alike and determine that UAV number 2 covers targets 2 **AND** 3 without any communication.

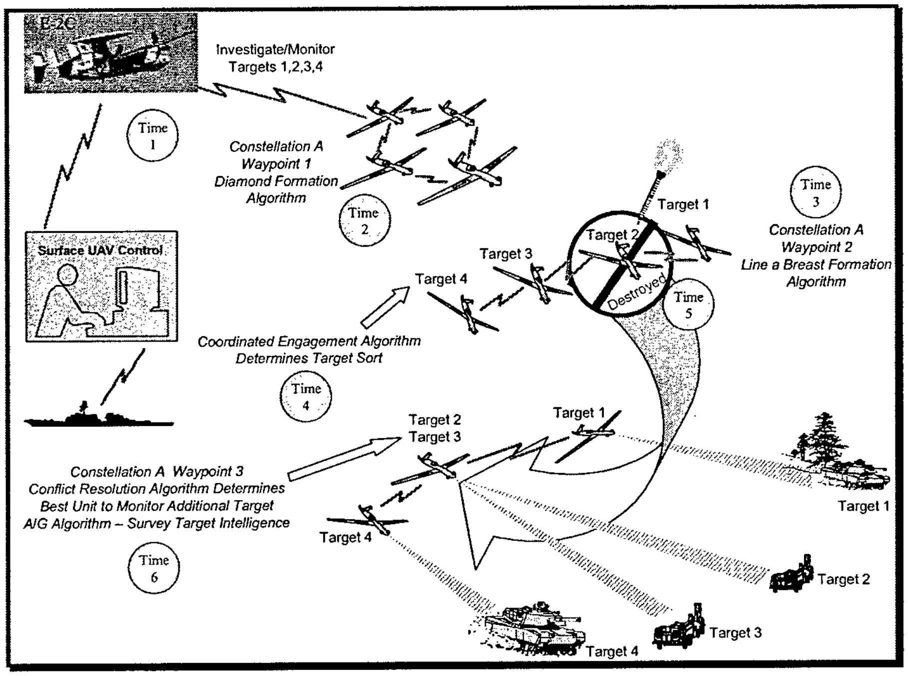

Figure 9 Concept of Algorithm Application to UAV Command and Control

To achieve this, three issues are considered: flight control, group behavior, and tactical mission. Results of understanding the limitations of control structures and data requirements will enable designers of autonomous systems. Well-designed plans have reserve capacity to achieve other missions, and situations can change significantly requiring deviations from the plan. The keys to all of these resource management issues are the timeliness and cooperative nature of the assets available. "Cooperative" includes taking advantage of matching the correct sensor platform with the target condition and optimization of platform routing, while timely includes doing so with sufficient swiftness to allow more effective targeting. A timely and cooperative assignment would have minimum impact, and achieve a high probability of mission success.

Other Applications: We move now from *Autonomous Vehicle Control* for a brief discussion of research that DSR is investigating in *System Control* using neural networks. A practical application being investigated for our neural networks is dynamic resource allocations. This is especially important involving Intelligence, Surveillance and Reconnaissance (ISR) to support engagement of enemy targets. One of the most stressing problems facing battle managers is the need to be able to quickly, efficiently and dynamically re-task and reallocate ISR assets in response to new information or changing battlefield conditions. This could be particularly true for mobile targets such as TEL missile launchers using "shoot and scoot" tactics that may only expose themselves for short periods of time before moving back into hide sites, or as an example for the Navy, a submarine periscopes that have broken the water surface prior to a missile launch. They both provide an obvious, but fleeting signature and location, and unless the battle manager can rapidly reposition ISR assets to the launch area, the target will be lost. Battle managers require the ability to quickly assess the unified ISR plan to determine if there is the appropriate capability (sensor/asset) available, if it can be retasked or diverted to the new mission in time, with the ability to determine impact on the unified plan. Management of the theater's ISR assets involves the positioning of platforms to obtain coverage, matching of sensor type (e.g. radar, or EO) to target signature or ISR task (e.g. tracking or ID), and determining the impact of the retasking on the overall theater ISR plan. Depending on the threat priority assigned to the fleeting target and the impact to the overall plan, a small subset of the sensor assets might be Apportioned, perform a Quick Look task, and Return to primary mission assignment (AQLR). Figure 10 describes the problem today, and with more efficient managing of assets utilizing DSR software techniques.

The elements controlled might consist of:

ISR Assets	Sensors	Mission, Task Or Function Assignments
▪ Joint Stars, ▪ U2, ▪ Rivet Joint, ▪ UAVs, (Global Hawk, Predator) ▪ Discoverer II	▪ Satellite (Space) ▪ GMTI Radar, ▪ SAR, ▪ EO/IR, ▪ SIGINT ▪ COMINT ▪ FOPEN Radar ▪ HRR ▪ UGS	▪ Detecting ▪ Classifying ▪ Tracking ground targets ▪ Battle Damage Assessment

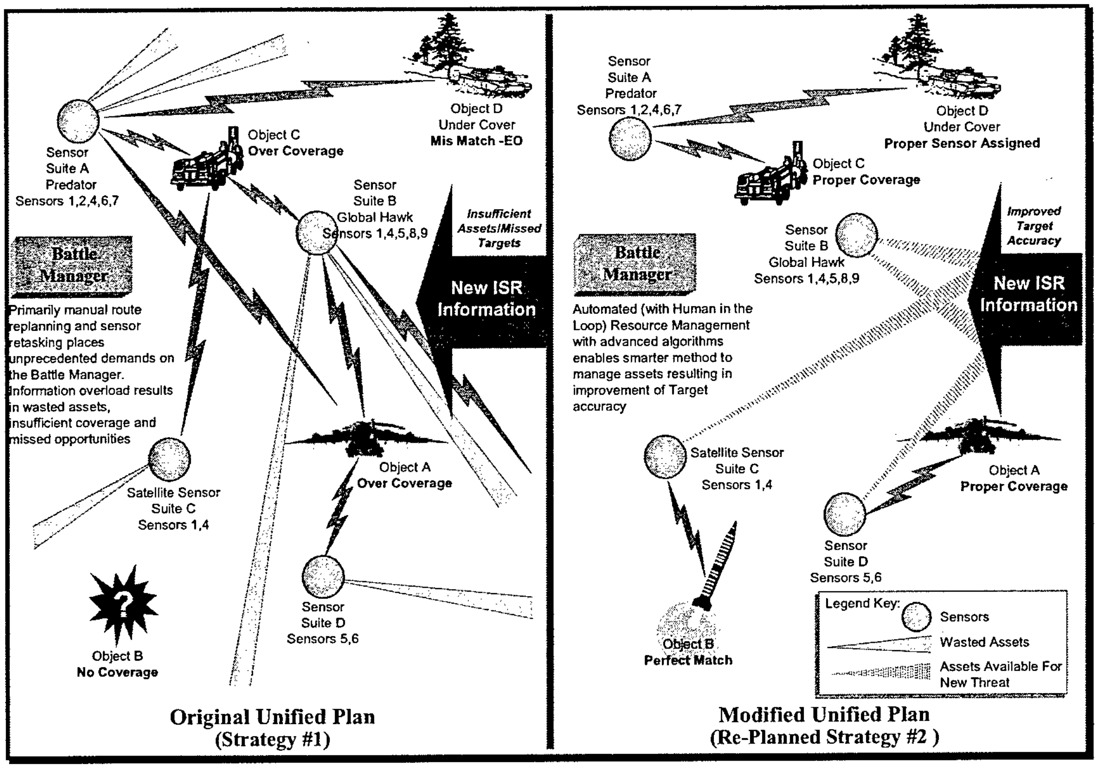

Figure 10 Effect of Asset Management with DSR Software Techniques

As postulated above, optimizing the employment (which combinations, their coordination and deconfliction) of these elements is the critical uncontrollable factor. It is unreasonable to expect a human operator to be familiar with this many sensors and platforms, and to track their positioning and sensor operating mode, as well as being able to detect small nuances in target behavior that indicates that that track is a target of interest. When one adds to this problem the possibility of hundreds of thousands of targets, over thousands of square miles in which the only discrete target behavior is its disappearance as it moves into a "deep hide condition" there could be an expectation that this problem can only be solved in a special facility. By combining our artificial intelligence neural network technologies with techniques that employ the use of interactive multi-model (IMM) Kalman filters embedded in a Bayesian network as a tightly coupled contact tracker and recognition algorithm with an operator/mission planner situational awareness capability to discriminate information and be opportunistic in the collection of data, especially for targets which may emerge for brief periods of time.